

(19)



Europäisches Patentamt
European Patent Office
Office européen des brevets



(11)

EP 1 077 405 A2

(12)

EUROPEAN PATENT APPLICATION

(43) Date of publication:

21.02.2001 Bulletin 2001/08

(51) Int. Cl.⁷: **G06F 9/44**

(21) Application number: **00306865.7**

(22) Date of filing: **10.08.2000**

(84) Designated Contracting States:

**AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE**

Designated Extension States:

AL LT LV MK RO SI

(30) Priority: **17.08.1999 US 375602**

(71) Applicant:

**International Business Machines Corporation
Armonk, NY 10504 (US)**

(72) Inventors:

• **Audleman, Kevin Forbes
Winchester, Hampshire SO21 2JN (GB)**

• **McBride, Kevin Michael**

Winchester, Hampshire SO21 2JN (GB)

• **Patterson, Betty Joan**

Winchester, Hampshire SO21 2JN (GB)

• **Spain, Kelly Ann**

Winchester, Hampshire SO21 2JN (GB)

(74) Representative:

**Zerbi, Guido Maria
Intellectual Property Department,
IBM United Kingdom Ltd.,
Hursley Park
Winchester, Hampshire SO21 2JN (GB)**

(54) **Generating a graphical user interface from a command syntax for managing multiple computer systems as one computer system**

(57) A graphical user interface can be automatically generated from a command syntax for managing multiple computer systems as one computer system. The command syntax is represented using an eXtensible Markup Language (XML) document file, an XML schema file, and a text description file. The XML document file, XML schema file, and text description file are maintained on a server and downloaded to a client as required. A user interface program uses the XML document, XML schema, and text descriptions to generate a "Wizard" that comprises the graphical user interface, wherein the Wizard provides a series of step-by-step dialogs for assisting the operator in creating a command from the command syntax.

EP 1 077 405 A2

Description

[0001] The present invention generally relates to the management of computer-implemented systems, and in particular, to a system and method for generating a graphical user interface from a command syntax for managing multiple computer systems as one computer system.

[0002] The management of computer systems has become increasingly more complex. This results from the complex hardware and software environments found in modern-day computer systems. For example, it is not uncommon to manage a large number of computer systems in a network, wherein the systems perform different tasks and run differing versions of software, wherein the systems are geographically dispersed, and wherein multiple operations consoles are required. As computer systems and networks grow in complexity, so too do the operations consoles and their user interfaces. Yet, it is not desirable to have a separate operations console for each computer system, nor is desirable to maintain a different client interface for each of the operations consoles.

[0003] Accordingly, the invention provides a method for generating a graphical user interface from a command syntax, for managing at least one computer system, comprising:

(a) processing at least one eXtensible Markup Language (XML) file at a user interface program executed by a computer, wherein the XML file represents a command syntax for managing the at least one computer system; and

(b) generating at least one dialog from the XML file for display to an operator, wherein the at least one dialog is used to create at least one command in the user interface program for managing the at least one computer system.

[0004] In a further aspect, a computer program produce is provided comprising computer program code stored on a computer readable storage medium, which when executed on a computer performs the method of the invention.

[0005] In a yet further aspect, the invention provides a system for generating a graphical user interface from a command syntax for managing at least one computer system, comprising:

(a) a computer; and

(b) a user interface program, executed by the computer, for processing at least one eXtensible Markup Language (XML) file at a user interface program executed by a computer, wherein the XML file represents a command syntax for managing the at least one computer system, and for generating at least one dialog from the XML file for display to an operator, wherein the at least one dialog is used to create at least one command in the user interface program for managing the at least one computer system.

[0006] Preferably a system is provided that can manage multiple computers and that has a client interface that is easy to maintain. The client interface reduces the complexity of managing computer systems, and does not require operators to remember a large repertoire of commands and command syntax.

[0007] According to a preferred embodiment the command syntax is represented using an eXtensible Markup Language (XML) document file, an XML schema file, and a text description file. The XML document file, XML schema file, and text description file are maintained on a server and downloaded to a client as required. A user interface program uses the XML document, XML schema, and text descriptions to generate a "Wizard" that comprises the graphical user interface, wherein the Wizard provides a series of step-by-step dialogs for assisting the operator in creating a command from the command syntax.

[0008] Preferably when the command syntax changes (for example, when a new function is added), updates are only needed at the server, thereby reducing the maintenance cost of clients. In addition, according to the preferred embodiment all clients are synchronized with the server and have the most up to date version of the command syntax. Moreover, generating the graphical user interface from the command syntax greatly reduces the amount of programming effort involved in creating the user interface. Finally, the command syntax is easier to learn, because the Wizard displays a step-by-step series of dialogs for guiding the operator in the use of the command syntax.

[0009] In the preferred embodiment, the server maintains control of the command syntax in order to support multiple versions thereof. Moreover, the client is preferably not sensitive to individual versions of the command syntax or it would require an update every time a command was changed. Finally, the tagging rules for the XML allows the user interface program to generate customized Wizards for the command syntax.

[0010] A preferred embodiment of the present invention will now be described in detail, by way of example only, and with reference to the following drawings:

FIG. 1 schematically illustrates the environment of a preferred embodiment of the present invention;

FIG. 2 is a flowchart that illustrates the logic of constructing the extensible Markup Language (XML) files according to a preferred embodiment of the present invention;

FIG. 3 is a flowchart that illustrates the logic of processing the XML files according to a preferred embodiment of the present invention; and

FIGS. 4A, 4B, 4C, 4D, 4E, 4F, and 4G illustrate a graphical user interface generated from the XML files according to a preferred embodiment of the present invention.

Please note, reference numerals represent corresponding parts throughout.

Hardware and Software Environment

[0011] FIG. 1 schematically illustrates the hardware and software environment of the preferred embodiment of the present invention including a distributed computer system 100 using a network 102 to connect one or more clients 104 to a server 106 having one or more data storage devices 108. The network 102 may comprise networks such as LANs, WANs, SNA networks, and the Internet. A typical combination of resources includes clients 104 that are implemented on personal computers or workstations, and servers 106 that are implemented on personal computers, workstations, minicomputers, or mainframes.

[0012] The server 106 is a sysplex that executes one or more copies of a Command Processor (CP) 108, which may comprise any number of systems, and any number of different types of systems. The multiple CP's 108 may act individually or as a single system in providing services and applications, but each CP 108 is managed separately, i.e., a separate command is sent to each CP 108 and each response from a CP 108 is viewed separately. As the number of CP's 108 increases, the manageability of the server 106 becomes more complex.

[0013] The server 106 also executes one or more Command Servers (CS) 110 that communicate with all of the CP's 108 and the clients 104 for providing operator control of the CP's 108. The CS 110 maintains a collection of related XML files 112 that represent the command syntax for the CP's 108. The XML files 112 store the command syntax as a separately replaceable unit.

[0014] The client 104 executes a user interface program 114 that interacts with the CS 110 to provide an operator with control over the CP's 108. The user interface program 114 receives a copy of the XML files 112 from the CS 110, wherein the XML files 112 represent a command syntax of the highest active release level of the command set for the CP's 108. The user interface program 114 processes the XML files 112 to identify resource types, verbs, and keywords. The user interface program 114 then dynamically displays a "Wizard" comprising a step-by-step series of dialogs that guide the operator through the command syntax, and constructs one or more commands based on the operator selections from the dialogs.

[0015] The CS 110 receives the commands from the user interface program 114, and submits the commands to one or more (possibly all) of the CP's 108. The CS 110 then receives responses from the CP's 108 and transmits the responses to the user interface program 114. For example, in response to operator commands, the CS 110 may provide the operator with a list of active CP's 108 executing in the system 100 and/or status notifications from the CP's 108.

[0016] Preferably, these components 108, 110, 112, and 114 all comprise logic and/or data that is embodied in or retrievable from a computer-readable device, medium, or carrier, e.g., a data storage device, a data communications device, a remote computer or device coupled to the computer via a data communications device, etc. Moreover, this logic and/or data, when read, executed, and/or interpreted by the computer, causes the computer to perform the steps necessary to implement and/or use a preferred embodiment of the present invention.

[0017] Thus, the present invention may, for example, be implemented as a method, apparatus, or computer program product using standard programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof. The term "article of manufacture" (or alternatively, "computer program product") as used herein is intended to encompass logic and/or data accessible from any computer-readable device, carrier, or media.

[0018] Those skilled in the art will recognize that any combination of the above components, or any number of different components, including different logic, data, different peripherals, and different devices, may be used to implement the present invention, so long as similar functions are performed thereby. Specifically, those skilled in the art will recognize that the present invention may be applied to any database, associated database management system, or peripheral device.

Command Syntax

[0019] In the preferred embodiment, the general format of the command syntax for the CP 108 comprises the following:

VERB	RESOURCETYPE	KEYWORD	(PARAMETER)
------	--------------	---------	---------------

Each of these elements is described below:

- 5 VERB - The verb is the first element and identifies the action to be taken. The verb can be abbreviated.
 RESOURCETYPE - The resource type is the second element and identifies the type of resource acted upon by the command. The resource may comprise, for example, transactions, databases, terminals, areas, users, etc. The resource type may support a synonym.
 10 KEYWORD - Keywords are optional elements, depending upon the specific command. The keyword may identify resources of one or more CP's 108. The keyword may also identify a function of the command.
 PARAMETER - A parameter identifies a defined or created value or resource, such as a database, terminal, area, user, etc. Parameters in commands are preferably replaced with values. Multiple parameters are separated by a comma within the parentheses.

- 15 Of course, those skilled in the art will recognize that other command syntax may be used, as could other elements.

XML Files

- 20 [0020] In the preferred embodiment, the XML files 112 provide a tag language representation of the command syntax, and include an XML document file, an XML schema file, and a text representation file. Examples of these files can be found in the Appendices attached hereto, wherein Appendix I is an exemplary XML document file (CommandSyntax.XML), Appendix II is an exemplary XML schema file (CommandSyntax.DTD), and Appendix III is an exemplary text description file (CommandText.TAG).

- 25 [0021] XML describes a class of data objects and partially describes the behavior of computer programs that process these objects. XML is a restricted form of SGML (Standard Generalized Markup Language), as defined in ISO 8879. The specification for XML can be found at the URL: <http://www.w3.org/TR/REC-xml>.

- [0022] An XML schema specifies constraints on the structures and types of elements in an XML document. The basic schema for XML is the DTD (Document Type Definition), although other XML schema definitions are also being developed, such as DCD (Document Content Definition), XSchema, etc. Information concerning DTD and DCD can be found at the URL: <http://www.w3.org/>.

- 30 [0023] Preferably, a text description file provides the text descriptions that are used by the user interface program to construct the dialogs of the Wizard. The text descriptions are represented by tags that comprise concatenations of the nested XML tag names in the XML document.

- 35 [0024] The user interface program 114 of the preferred embodiment incorporates an XML processor that is used to read the XML files 112 and provide access to their content and structure. The XML specification located at the URL noted above describes how an XML processor typically behaves in terms of how it processes XML data.

XML Document File

- 40 [0025] As shown in Appendix I, the XML document file includes an (?xml) version identifier, an (!DOCTYPE root SYSTEM "CommandSyntax.dtd") identifier of the XML schema file (e.g., .dtd file), and opening and closing (root) and /root tags. The remaining tags in the XML document file are described below in conjunction with the XML schema file.

- 45 [0026] The command syntax of the XML document file comprises the command syntax for a database management system (DBMS), and more specifically, the IMS™ DBMS sold by IBM Corporation, the assignee of the present invention.

XML Schema File

- 50 [0027] Each tag in the XML document file has a set of rules that must be followed, which are defined in the XML schema file. As shown in Appendix II, most of the tags have descriptive names that indicate their function. Following is a description of the tags and their attributes in the preferred embodiment:

- 55 root - Used as a global tag that provides a wrapper for all the other tags. Attributes: none.
 resource - Used to define a resource. Attributes: name - REQUIRED (specifies the name of the resource).
 verb - Used to define a verb. Attributes: name - REQUIRED (specifies the name of the verb).
 keyword - Used to define a keyword. Attributes: name - REQUIRED (specifies the name of the keyword); required - OPTIONAL (required parameter for command? Yes or No. Default is No).
 No. Default is No).

list - Used within (keyword) to define a list of options. Attributes: name - OPTIONAL (specifies the name of the group); default - OPTIONAL (default value for var (see later)); multiple - OPTIONAL (does var accept multiple values? Yes or No. Default is No.); and goto - OPTIONAL (point to another keyword to be enabled).

compare - Used within (keyword) to define a comparison between an operator and a variable. Attributes: name - REQUIRED (name of comparison).

option - Used within (list) to make a radio button or checkbox. Attributes: name - REQUIRED (name of keyword this is pointing to);

goto - OPTIONAL (point to another keyword to be enabled, wherein the goto attribute of the (list) and (option) tags means that the specified keyword is applicable only when that list or option is selected.)

var - Used within (keyword) and (list) to define a variable. Attributes: name - REQUIRED (name of variable); wildcard - OPTIONAL (does this var support wildcards? Yes or No. Default is No.); default - OPTIONAL (default value for var.); and multiple - OPTIONAL (does var accept multiple values? Yes or No. Default is No.).

operator - Used within (compare) to define an operator for the list of operators. Attributes: keyword - REQUIRED (name of keyword this goto points to).

Text Description File

[0028] As shown in Appendix III, the tags in the text description file are concatenated versions of tags used in the XML document file. By traversing the XML document file and concatenating the tag names therein, the user interface program 114 can identify the tags used in the text description file and thereby access the text to be displayed by the Wizard. Generally, this text is in HTML (HyperText Markup Language) format, although other formats may be used as well.

[0029] The text description file is used for the following reasons:

(1) To provide enhanced output for the operator. In the XML document file, commands are often represented by mnemonics that are recognized by the CP 108, but which may or may not be completely understandable by the operator.

(2) To allow for internationalization or localization of the command syntax. The user interface program 114 may be used in multiple countries, and language support should be provided for each country. Since the XML document and XML schema files cannot be changed (since their format is dictated by the CP 108), the text description file provides a mechanism whereby the Wizard output from the user interface program 114 is understandable by the operator.

(3) To simplify the XML document file. The use of the text description file avoids the inclusion of confusing text descriptions within the XML document file.

Logic of the Preferred Embodiment

[0030] Flowcharts which illustrate the logic according to the preferred embodiment of the present invention are shown in FIGS. 2-3. Those skilled in the art will recognize that this logic is provided for illustrative purposes only and that different logic may be used to accomplish the same results.

[0031] FIG. 2 is a flowchart that illustrates the logic of encoding the XML files 112 according to the preferred embodiment of the present invention.

[0032] Block 200 represents the XML schema file being defined and then stored on the server 106. Preferably, this structure should not change as the command syntax changes.

[0033] Block 202 represents the XML document file being defined and then stored on the server 106. The XML document file is defined. Preferably, this structure will change as the command syntax changes. The Block includes the following steps:

a. As noted above, the XML document file includes an (?xml) version identifier, an (DOCTYPE root SYSTEM "CommandSyntax.dtd") identifier of the XML schema file (e.g., .dtd file), and opening and closing (root) and /root) tags.

b. The resources are defined (TRAN, OLC, ...) with (resource) tags. For each resource, its verbs are defined (QUERY, UPDATE, ...) with (verb) tags. For each verb, its keywords are defined with (keyword) tags. A keyword can include either a variable (a (var) tag), a list of options ((list) and (option) tags), or a list of operators for comparison ((compare) and (operator) tags).

c. Each command is uniquely identified by a (resource) tag followed by a (verb) tag. Each keyword in the command

has its own (keyword) tag, and the type of information a keyword contains can be ascertained by its children. If the keyword contains a (var), then the operator will be prompted to input a variable name.

- 5 [0034] Block 204 represents the text description file being defined and then stored on the server 106. Preferably, this structure will change as the command syntax changes.
- [0035] FIG. 3 is a flowchart that illustrates the logic of processing the XML files 112 according to the preferred embodiment of the present invention.
- [0036] Block 300 represents the user interface program 114 receiving the XML files 112 from the server 106.
- 10 [0037] Block 302 represents the user interface program 114, using the XML schema file as a template, traversing through the tags in the XML document file to gather information from each tag.
- [0038] Block 304 represents the user interface program 114 constructing a "Wizard" for the XML document file using the text description file, wherein the Wizard displays a series of steps to the operator. Each of the steps displayed by the Wizard corresponds to a (keyword) element in the XML document file. The text description file is used to define the display elements for the Wizard. In the example XML document file provided above, the Wizard would have seven
- 15 steps as described further below.
- [0039] Block 306 represents the user interface program 114 displaying the command as it has been built via the operator selections of the Wizard, wherein a button or other mechanism is provided for the operator to submit the command to one or more of the CP's 108.
- 20 [0040] Block 308 represents the user interface program 114 displaying any responses received from the CP's 108 for the operator.

Graphical User Interface

- 25 [0041] FIGS. 4A, 4B, 4C, 4D, 4E, 4F, and 4G illustrate the graphical user interface generated from the XML files 112 according to the preferred embodiment of the present invention. These particular "snapshots" of the graphical user interface correspond to the Wizard generated for the example XML files 112 described in the Appendices. Of course, those skilled in the art will recognize that this graphical user interface is provided for illustrative purposes only and that a different graphical user interface may be used to accomplish the same results.
- 30 [0042] Specifically, the Wizard shown in FIGS. 4A, 4B, 4C, 4D, 4E, 4F, and 4G corresponds to the following portion of Appendix I:

```
<resource name="TRAN">
```

```
35      <verb name="QUERY">
```

```
      <keyword name="NAME" required="YES">
```

```
40          <var name="TRAN_NAME" wildcard="YES"
              multiple="YES" default=""/>
```

```
45
```

```
50
```

```
55
```

```
</keyword>
```

```
<keyword name="CLASS">
```

```
<var name="CLASS_NUMBER" multiple="YES"/>
```

```
</keyword>
```

```
<keyword name="STATUS">
```

```
<list>
```

```
<option name="DEFAULT"/>
```

```
<list name="LIST" multiple="YES">
```

```
<option name="ACT"/>
```

```
<option name="BAL"/>
```

```
<option name="INACT"/>
```

```
<option name="IOPREV"/>
```

```
<option name="LCK"/>
```

```
<option name="OLC"/>
```

```
<option name="PSTO"/>
```

```
<option name="PUR"/>
```

```
<option name="QERR"/>
```

```
<option name="SPND"/>
```

```
<option name="STO"/>
```

```
<option name="TRA"/>
```

```
<option name="USTO"/>
```

```
</list>
```

```
</list>
```

```
</keyword>
```

```
<keyword name="QCNT">
```

```
<compare name="NUMBER">
```

```
<operator name="GT"/>
```

```
<operator name="LT"/>
```

```
<operator name="EQ"/>
```

```
<operator name="GE"/>
```

```
<operator name="LE"/>
```

```
<operator name="NE"/>
```

```
</compare>
```

```
</keyword>
```

```
<keyword name="SHOW">
```

```
<list>
```

```
<option name="DEFAULT"/>
```

```

5      <option name="ALL"/>
      <option name="GLOBAL"/>
      <option name="LOCAL"/>
      <list name="LIST" multiple="YES">
10         <option name="CLS"/>
         <option name="CP"/>
         <option name="LCT"/>
         <option name="LP"/>
         <option name="NP"/>
15         <option name="PARLM"/>
         <option name="PLCT"/>
         <option name="PSBNAME"/>
         <option name="QCNT"/>
20         <option name="RC"/>
         <option name="SEGNO"/>
         <option name="SEGSZ"/>
         <option name="STATUS"/>
25     </list>
    </list>
    </keyword>
30
    <keyword name="ROUTE">
        <var name="IMS" wildcard="YES" multiple="YES"
35        default="*" />
    </keyword>
40
    </verb>
    </resource>

```

45 [0043] In this example, the CP's 108 each comprise a database management system (DBMS), and more specifically, the IMS™ DBMS sold by IBM Corporation, the assignee of the present invention. Thus, as noted above, the command syntax of the XML document file comprises the command syntax for an IMS™ DBMS.

50 [0044] FIG. 4A illustrates step 1 of 7 of the Wizard, which prompts the operator to name the transactions to display. This page of the Wizard lists the names of the transactions that will be displayed. The operator types a transaction name in the entry field and presses Enter to add it to the list. The operator can also select previous transactions from the drop-down list and use the wildcard (*) character to display all or filter transactions.

55 [0045] FIG. 4B illustrates step 2 of 7 of the Wizard, which prompts the operator to specify the class number of transactions to display. This page of the Wizard lists the class numbers of the transactions that will be displayed. The operator types a class number in the entry field and presses Enter to add it to the list. The operator can also select previous class numbers from the drop-down list.

[0046] FIG. 4C illustrates step 3 of 7 of the Wizard, which prompts the operator to select a status of the transactions to display. This page of the Wizard lists transaction status types that the operator can select. The select will apply to the

transactions that the operator specified in the previous Name and Class steps (i.e., steps 1 and 2).

[0047] FIG. 4D illustrates step 4 of 7 of the Wizard, which prompts the operator to specify the queue count of transactions to display. This page of the Wizard is used to filter transactions by queue count. The operator chooses the comparison operator from the list and then types a number in the entry field.

5 **[0048]** FIG. 4E illustrates step 5 of 7 of the Wizard, which prompts the operator to select the transaction output field to display. This page of the Wizard is used to specify the transaction output fields to be displayed and overrides the default transaction output fields.

[0049] FIG. 4F illustrates step 6 of 7 of the Wizard, which prompts the operator to specify the CP 108 that will receive this command. This page of the Wizard is used to complete the operation and submit the command.

10 **[0050]** FIG. 4G illustrates step 7 of 7 of the Wizard, which prompts the operator to review the command generated for the specified CP 108. This page of the Wizard is used to specify when the generated command will be submitted to the specified CP 108.

15 **[0051]** Those skilled in the art will recognize that different database management systems, operating environments, etc., could be substituted for those described herein. Those skilled in the art will recognize that the present invention could be used in any type of computer system, and need not be limited to a client-server architecture. Those skilled in the art will recognize that the present invention could be used with many types of computer systems, and need not be limited to the examples described herein.

20

25

30

35

40

45

50

55

Appendix I

5 <?xml version="1.0"?>

10 <!-- *****
 **
 ** This file contains sample command syntax in XML
 ** for the following resources:
 ** LTERM resource, QUERY and UPDATE verbs
 ** TRAN resource, QUERY and UPDATE verbs
 **
 15 ***** -->

<!DOCTYPE root SYSTEM "CommandSyntax.dtd">

20 <root>

<resource name="LTERM">

25 <verb name="QUERY">

<keyword name="NAME" required="YES">

30 <var name="LTERM_NAME" wildcard="YES" multiple="YES" default=""/>

</keyword>

<keyword name="STATUS">

35 <list>

<option name="DEFAULT"/>

<list name="LIST" multiple="YES">

<option name="LCK"/>

40 <option name="PSTO"/>

<option name="PUR"/>

<option name="QERR"/>

<option name="QLCK"/>

45 <option name="STATIC"/>

<option name="STO"/>

</list>

</list>

50 </keyword>

<keyword name="QCNT">

55

```

5      <compare name="NUMBER">
        <operator name="GT"/>
        <operator name="LT"/>
        <operator name="EQ"/>
        <operator name="GE"/>
        <operator name="LE"/>
10     <operator name="NE"/>
      </compare>
    </keyword>

15    <keyword name="SHOW">
      <list>
        <option name="DEFAULT"/>
        <option name="ALL"/>
20     <option name="GLOBAL"/>
        <option name="LOCAL"/>
        <list name="LIST" multiple="YES">
          <option name="NODE"/>
25     <option name="QCNT"/>
          <option name="STATUS"/>
          <option name="USER"/>
        </list>
30    </list>
    </keyword>

    <keyword name="ROUTE">
35     <var name="IMS" wildcard="YES" multiple="YES" default=""/>
    </keyword>

    </verb>

40    <verb name="UPDATE">

      <keyword name="NAME" required="YES">
45     <var name="LTERM_NAME" wildcard="YES" multiple="YES"/>
      </keyword>

      <keyword name="START">
50     <list>
        <option name="DEFAULT"/>
        <option name="ALL"/>
55

```

```

5      <list name="LIST" multiple="YES">
        <option name="IN"/>
        <option name="OUT"/>
        <option name="QUE"/>
      </list>
    </list>
10   </keyword>

    <keyword name="STOP">
      <list>
15      <option name="DEFAULT"/>
        <option name="ALL"/>
        <list name="LIST" multiple="YES">
          <option name="IN"/>
          <option name="OUT"/>
20      <option name="QUE"/>
        </list>
      </list>
25   </keyword>

    <keyword name="ROUTE">
      <var name="IMS" wildcard="YES" multiple="YES" default=""/>
30   </keyword>

  </verb>

35 </resource>

40 </resource>

  <resource name="TRAN">

45   <verb name="QUERY">

      <keyword name="NAME" required="YES">
        <var name="TRAN_NAME" wildcard="YES" multiple="YES" default=""/>
50   </keyword>

      <keyword name="CLASS">
55

```

```

    <var name="CLASS_NUMBER" multiple="YES"/>
</keyword>

```

5

```

<keyword name="STATUS">

```

```

  <list>

```

```

    <option name="DEFAULT"/>

```

```

    <list name="LIST" multiple="YES">

```

10

```

      <option name="ACT"/>

```

```

      <option name="BAL"/>

```

```

      <option name="INACT"/>

```

```

      <option name="IOPREV"/>

```

15

```

      <option name="LCK"/>

```

```

      <option name="OLC"/>

```

```

      <option name="PSTO"/>

```

```

      <option name="PUR"/>

```

20

```

      <option name="QERR"/>

```

```

      <option name="SPND"/>

```

```

      <option name="STO"/>

```

25

```

      <option name="TRA"/>

```

```

      <option name="USTO"/>

```

```

    </list>

```

```

  </list>

```

```

</keyword>

```

30

```

<keyword name="QCNT">

```

```

  <compare name="NUMBER">

```

```

    <operator name="GT"/>

```

35

```

    <operator name="LT"/>

```

```

    <operator name="EQ"/>

```

```

    <operator name="GE"/>

```

40

```

    <operator name="LE"/>

```

```

    <operator name="NE"/>

```

```

  </compare>

```

```

</keyword>

```

45

```

<keyword name="SHOW">

```

```

  <list>

```

```

    <option name="DEFAULT"/>

```

```

    <option name="ALL"/>

```

50

```

    <option name="GLOBAL"/>

```

```

    <option name="LOCAL"/>

```

55

```

5      <list name="LIST" multiple="YES">
        <option name="CLS"/>
        <option name="CP"/>
        <option name="LCT"/>
        <option name="LP"/>
        <option name="NP"/>
10      <option name="PARLM"/>
        <option name="PLCT"/>
        <option name="PSBNAME"/>
        <option name="QCNT"/>
15      <option name="RC"/>
        <option name="SEGNO"/>
        <option name="SEGSZ"/>
        <option name="STATUS"/>
20      </list>
    </list>
</keyword>

25      <keyword name="ROUTE">
        <var name="IMS" wildcard="YES" multiple="YES" default=""/>
    </keyword>

30      </verb>

    <verb name="UPDATE">

35      <keyword name="NAME" required="YES">
        <var name="TRAN_NAME" wildcard="YES" multiple="YES"/>
    </keyword>

40      <keyword name="CLASS">
        <var name="CLASS_NUMBER" multiple="YES"/>
    </keyword>

45      <keyword name="START">
        <list>
            <option name="DEFAULT"/>
            <option name="ALL"/>
50      <list name="LIST" multiple="YES">
            <option name="IN"/>
            <option name="OUT"/>
55

```

```

5      <option name="QUE"/>
      <option name="SCHED"/>
      <option name="SPNDQ"/>
      <option name="TRC"/>
      </list>
    </list>
10  </keyword>

    <keyword name="STOP">
      <list>
15      <option name="DEFAULT"/>
      <option name="ALL"/>
      <list name="LIST" multiple="YES">
        <option name="IN"/>
20      <option name="OUT"/>
        <option name="QUE"/>
        <option name="SCHED"/>
        <option name="TRC"/>
25      </list>
      </list>
    </keyword>

30  <keyword name="SET">
      <list name="LIST" multiple="YES">
        <var name="CLASS"/>
        <var name="CPRI"/>
35      <var name="LMCT"/>
        <var name="LPRI"/>
        <var name="MAXRGNS"/>
        <var name="NPRI"/>
40      <var name="PARLIM"/>
        <var name="PLMCT"/>
        <var name="SEGNO"/>
        <var name="SEGSIZE"/>
45      </list>
    </keyword>

50  <keyword name="ROUTE">
      <var name="IMS" wildcard="YES" multiple="YES" default=""/>
    </keyword>

```

</verb>

5

</resource>

</root>

10

15

20

25

30

35

40

45

50

55

Appendix II

```

5  <!-- *****
   **
   ** This file defines the command syntax DTD
   **
10 ***** -->

   <!ELEMENT root ( resource* )>
   <!ELEMENT resource ( verb* )>
15 <!ELEMENT verb ( keyword* )>
   <!ELEMENT keyword ( list* | compare | var )>
   <!ELEMENT list ( list | var | option )*>
20 <!ELEMENT compare ( operator* )>
   <!ELEMENT option EMPTY>
   <!ELEMENT var EMPTY>
25 <!ELEMENT operator EMPTY>

   <!ATTLIST resource name CDATA #REQUIRED>
   <!ATTLIST verb name CDATA #REQUIRED>
30 <!ATTLIST keyword name CDATA #REQUIRED
      required CDATA #IMPLIED>
   <!ATTLIST var name CDATA #REQUIRED
      wildcard CDATA #IMPLIED
35      default CDATA #IMPLIED
      multiple CDATA #IMPLIED>
   <!ATTLIST option name CDATA #REQUIRED
40      goto CDATA #IMPLIED>
   <!ATTLIST list name CDATA #IMPLIED
      default CDATA #IMPLIED
      multiple CDATA #IMPLIED
45      goto CDATA #IMPLIED>
   <!ATTLIST compare name CDATA #REQUIRED>
   <!ATTLIST operator name CDATA #REQUIRED>
50
55

```

Appendix III

```

*****
5  **
  ** This file contains sample text strings used in the command syntax
  ** for the following resources:
10  ** LTERM resource, QUERY and UPDATE verbs
  ** TRAN resource, QUERY and UPDATE verbs
  **
*****

15  LTERM_NAME
    "Logical Terminal"

20  LTERM_QUERY_NAME
    "Query"

    LTERM_QUERY_NAME_NAME
    "Name"

25  LTERM_QUERY_NAME_STEP_TITLE
    "Name logical terminals to display"
    LTERM_QUERY_NAME_STEP_TEXT
    "This page lists the names of the logical terminals that you will
30  display. Type a logical terminal name in the entry field and
    press Enter to add it to the list. You can also select
    previous logical terminals from the drop-down list and use the
    wildcard (*) character to display all or filter logical
35  terminals. For example, ABC* will update all logical terminals
    starting with ABC."
    LTERM_QUERY_NAME_CONTROLS_TITLE
    "Logical terminal names (*)"

40  LTERM_QUERY_STATUS_NAME
    "Status"

    LTERM_QUERY_STATUS_STEP_TITLE
    "Select status of logical terminals to display"
45  LTERM_QUERY_STATUS_STEP_TEXT
    "This page lists logical terminal status types that you can
    select. The select will apply to the logical terminals that
50  you specified in the Name and Class steps. To see a detailed
    explanation of an option, move the cursor to that option."

```

```

LTERM_QUERY_STATUS_CONTROLS_TITLE
  "Options"
5 LTERM_QUERY_STATUS_CONTROLS_HELP
  "Option help"
LTERM_QUERY_STATUS_DEFAULT_NAME
  "Default - All"
10 LTERM_QUERY_STATUS_DEFAULT_HELP
  "This is the default -- select all status"
LTERM_QUERY_STATUS_LIST_NAME
  "Select - List"
15 LTERM_QUERY_STATUS_LIST_HELP
  "Select the status options from the list."
LTERM_QUERY_STATUS_LIST_LCK_NAME
  "Lock"
20 LTERM_QUERY_STATUS_LIST_LCK_HELP
  "This is the help for Lock"
LTERM_QUERY_STATUS_LIST_PSTO_NAME
  "Process Stop"
25 LTERM_QUERY_STATUS_LIST_PSTO_HELP
  "This is the help for Process Stop"
LTERM_QUERY_STATUS_LIST_PUR_NAME
  "Purge"
30 LTERM_QUERY_STATUS_LIST_PUR_HELP
  "This is the help for Purge"
LTERM_QUERY_STATUS_LIST_QERR_NAME
  "Queue Error"
35 LTERM_QUERY_STATUS_LIST_QERR_HELP
  "This is the help for Queue Error"
LTERM_QUERY_STATUS_LIST_QLCK_NAME
  "Queue Lock"
40 LTERM_QUERY_STATUS_LIST_QLCK_HELP
  "This is the help for Queue Lock"
LTERM_QUERY_STATUS_LIST_STATIC_NAME
  "Static"
45 LTERM_QUERY_STATUS_LIST_STATIC_HELP
  "This is the help for Static"
LTERM_QUERY_STATUS_LIST_STO_NAME
  "Stop"
50 LTERM_QUERY_STATUS_LIST_STO_HELP
  "This is the help for Stop"

```

55

LTERM_QUERY_QCNT_NAME

"QCount"

LTERM_QUERY_QCNT_STEP_TITLE

"Specify the queue count of logical terminals to display"

LTERM_QUERY_QCNT_STEP_TEXT

"Use this page to filter logical terminals by queue count.
Choose the comparison operator from the list and then type a
number in the entry field. The following operators apply:

<TABLE>

<TR><TD>

GT > Greater than

LT < Less than

EQ = Equal to

</TD>

<TD>

GE >= Greater than or equal to

LE <= Less than or equal to

NE <> Not equal to

</TD>

</TABLE>"

LTERM_QUERY_QCNT_CONTROLS_TITLE

"Queue Count"

LTERM_QUERY_QCNT_OPERATOR

"Operator"

LTERM_QUERY_QCNT_NUMBER

"Number"

LTERM_QUERY_QCNT_NUMBER_GT_NAME

" > "

LTERM_QUERY_QCNT_NUMBER_LT_NAME

" < "

LTERM_QUERY_QCNT_NUMBER_EQ_NAME

" = "

LTERM_QUERY_QCNT_NUMBER_GE_NAME

" >= "

LTERM_QUERY_QCNT_NUMBER_LE_NAME

" <= "

LTERM_QUERY_QCNT_NUMBER_NE_NAME

" < > "

```

5      LTERM_QUERY_SHOW_NAME
        "Show"
      LTERM_QUERY_SHOW_STEP_TITLE
        "Select the logical terminal output field to display"
10     LTERM_QUERY_SHOW_STEP_TEXT
        "Specifies the logical terminal output fields to be
          displayed and overrides the default logical terminal
          output fields."
15     LTERM_QUERY_SHOW_CONTROLS_TITLE
        "Options"
      LTERM_QUERY_SHOW_CONTROLS_HELP
        "Option help"
20     LTERM_QUERY_SHOW_DEFAULT_NAME
        "Default - None"
      LTERM_QUERY_SHOW_DEFAULT_HELP
        "Show is an optional keyword. Select this to
          prevent SHOW from appearing in your command."
25     LTERM_QUERY_SHOW_ALL_NAME
        "All"
      LTERM_QUERY_SHOW_ALL_HELP
        "Selects all fields"
30     LTERM_QUERY_SHOW_GLOBAL_NAME
        "Global"
      LTERM_QUERY_SHOW_GLOBAL_HELP
        "This is the help for Show Global."
35     LTERM_QUERY_SHOW_LOCAL_NAME
        "Local"
      LTERM_QUERY_SHOW_LOCAL_HELP
        "This is the help for Show Local."
40     LTERM_QUERY_SHOW_LIST_NAME
        "Select - List"
      LTERM_QUERY_SHOW_LIST_HELP
        "Select from the list"
45     LTERM_QUERY_SHOW_LIST_NODE_NAME
        "NODE"
      LTERM_QUERY_SHOW_LIST_NODE_HELP
        "This is the help for Node"
50     LTERM_QUERY_SHOW_LIST_QCNT_NAME
        "QCNT"

```

55

LTERM_QUERY_SHOW_LIST_QCNT_HELP
 "Display the global queue count and the local queue count."
 5 LTERM_QUERY_SHOW_LIST_STATUS_NAME
 "STATUS"
 LTERM_QUERY_SHOW_LIST_STATUS_HELP
 "Global and local logical terminal status."
 10 LTERM_QUERY_SHOW_LIST_USER_NAME
 "User"
 LTERM_QUERY_SHOW_LIST_USER_HELP
 "This is the help for User"
 15
 LTERM_QUERY_ROUTE_NAME
 "Route"
 LTERM_QUERY_ROUTE_STEP_TITLE
 20 "Specify the systems that will receive this command"
 LTERM_QUERY_ROUTE_STEP_TEXT
 "This is the explanation of how to complete this step.
 It includes info on what the terms mean and how to use the
 25 controls and any available help."
 LTERM_QUERY_ROUTE_CONTROLS_TITLE
 "Modular unit id"
 30
 LTERM_UPDATE_NAME
 "Update"
 35
 LTERM_UPDATE_NAME_NAME
 "Name"
 LTERM_UPDATE_NAME_STEP_TITLE
 "Name logical terminals to update"
 40
 LTERM_UPDATE_NAME_STEP_TEXT
 "This page lists the names of the logical terminals that you will
 update. Type a logical terminal name in the entry field and
 45 press Enter to add it to the list. You can also select
 previous logical terminals from the drop-down list and use the
 wildcard (*) character to display all or filter logical
 terminals. For example, ABC* will update all logical terminals
 50 starting with ABC."
 LTERM_UPDATE_NAME_CONTROLS_TITLE
 "Transaction names (*)"

LTERM_UPDATE_START_NAME

"Start"

LTERM_UPDATE_START_STEP_TITLE

"Select logical terminal functions to stop"

LTERM_UPDATE_START_STEP_TEXT

"This page lists logical terminal functions that you can start.
The start will apply to the logical terminals that you selected
in the Name and Class steps. To see a detailed explanation
of an option, move the cursor to that option."

LTERM_UPDATE_START_CONTROLS_TITLE

"Options"

LTERM_UPDATE_START_CONTROLS_HELP

"Option help"

LTERM_UPDATE_START_DEFAULT_NAME

"Default - None"

LTERM_UPDATE_START_DEFAULT_HELP

"This is the default -- do not start any functions"

LTERM_UPDATE_START_ALL_NAME

"All"

LTERM_UPDATE_START_ALL_HELP

"Start all of the logical terminal functions."

LTERM_UPDATE_START_LIST_NAME

"Select - List"

LTERM_UPDATE_START_LIST_HELP

"Select the start options from the list."

LTERM_UPDATE_START_LIST_IN_NAME

"Input"

LTERM_UPDATE_START_LIST_IN_HELP

"This is the help for Input"

LTERM_UPDATE_START_LIST_OUT_NAME

"Output"

LTERM_UPDATE_START_LIST_OUT_HELP

"This is the help for Output"

LTERM_UPDATE_START_LIST_QUE_NAME

"Queueing"

LTERM_UPDATE_START_LIST_QUE_HELP

"This is the help for Queueing"

LTERM_UPDATE_STOP_NAME

"Stop"

LTERM_UPDATE_STOP_STEP_TITLE

"Select logical terminal functions to stop"

LTERM_UPDATE_STOP_STEP_TEXT

"This page lists logical terminal functions that you can stop.
The stop will apply to the logical terminals that you selected
in the Name and Class steps. To see a detailed explanation
of an option, move the cursor to that option."

LTERM_UPDATE_STOP_CONTROLS_TITLE

"Options"

LTERM_UPDATE_STOP_CONTROLS_HELP

"Option help"

LTERM_UPDATE_STOP_DEFAULT_NAME

"Default - None"

LTERM_UPDATE_STOP_DEFAULT_HELP

"This is the default -- do not stop any functions"

LTERM_UPDATE_STOP_ALL_NAME

"All"

LTERM_UPDATE_STOP_ALL_HELP

"Stop all of the logical terminal options."

LTERM_UPDATE_STOP_LIST_NAME

"Select - List"

LTERM_UPDATE_STOP_LIST_HELP

"Select the stop options from the list."

LTERM_UPDATE_STOP_LIST_IN_NAME

"Input"

LTERM_UPDATE_STOP_LIST_IN_HELP

"This is the help for Input"

LTERM_UPDATE_STOP_LIST_OUT_NAME

"Output"

LTERM_UPDATE_STOP_LIST_OUT_HELP

"This is the help for Output"

LTERM_UPDATE_STOP_LIST_QUE_NAME

"Queueing"

LTERM_UPDATE_STOP_LIST_QUE_HELP

"This is the help for Queueing"

LTERM_UPDATE_ROUTE_NAME

"Route"

LTERM_UPDATE_ROUTE_STEP_TITLE

"Specify the systems that will receive this command"

LTERM_UPDATE_ROUTE_STEP_TEXT

"This is the explanation of how to complete this step."

It includes info on what the terms mean and how to use the controls and any available help."

LTERM_UPDATE_ROUTE_CONTROLS_TITLE
"Modular unit id"

TRAN_NAME

"Transaction"

TRAN_QUERY_NAME

"Query"

TRAN_QUERY_NAME_NAME

"Name"

TRAN_QUERY_NAME_STEP_TITLE

"Name transactions to display"

TRAN_QUERY_NAME_STEP_TEXT

"This page lists the names of the transactions that you will display. Type a transaction name in the entry field and press Enter to add it to the list. You can also select previous transactions from the drop-down list and use the wildcard (*) character to display all or filter transactions. For example, ABC* will update all transactions starting with ABC."

TRAN_QUERY_NAME_CONTROLS_TITLE

"Transaction names (*)"

TRAN_QUERY_CLASS_NAME

"Class"

TRAN_QUERY_CLASS_STEP_TITLE

"Specify the class number of transactions to display"

TRAN_QUERY_CLASS_STEP_TEXT

"This page lists the class numbers of the transactions that you will display. Type a class number in the entry field and press Enter to add it to the list. You can also select previous class numbers from the drop-down list."

TRAN_QUERY_CLASS_CONTROLS_TITLE

"Class number"

TRAN_QUERY_STATUS_NAME
 "Status"
 5 TRAN_QUERY_STATUS_STEP_TITLE
 "Select status of transactions to display"
 TRAN_QUERY_STATUS_STEP_TEXT
 "This page lists transaction status types that you can select.
 10 The select will apply to the transactions that you specified
 in the Name and Class steps. To see a detailed explanation
 of an option, move the cursor to that option."
 TRAN_QUERY_STATUS_CONTROLS_TITLE
 15 "Options"
 TRAN_QUERY_STATUS_CONTROLS_HELP
 "Option help"
 TRAN_QUERY_STATUS_DEFAULT_NAME
 "Default - All"
 20 TRAN_QUERY_STATUS_DEFAULT_HELP
 "This is the default -- select all status."
 TRAN_QUERY_STATUS_LIST_NAME
 "Select - List"
 25 TRAN_QUERY_STATUS_LIST_HELP
 "Select the status options from the list."
 TRAN_QUERY_STATUS_LIST_ACT_NAME
 "Active"
 30 TRAN_QUERY_STATUS_LIST_ACT_HELP
 "This is the help for Active."
 TRAN_QUERY_STATUS_LIST_BAL_NAME
 "Balancing"
 35 TRAN_QUERY_STATUS_LIST_BAL_HELP
 "This is the help for Balancing"
 TRAN_QUERY_STATUS_LIST_INACT_NAME
 "Inactive"
 40 TRAN_QUERY_STATUS_LIST_INACT_HELP
 "This is the help for Inactive"
 TRAN_QUERY_STATUS_LIST_IOPREV_NAME
 "IO Prev"
 45 TRAN_QUERY_STATUS_LIST_IOPREV_HELP
 "This is the help for IO Prev"
 TRAN_QUERY_STATUS_LIST_LCK_NAME
 "Lock"
 50 TRAN_QUERY_STATUS_LIST_LCK_HELP
 "This is the help for Lock"

55

```

TRAN_QUERY_STATUS_LIST_OLC_NAME
    "Online Change"
5  TRAN_QUERY_STATUS_LIST_OLC_HELP
    "This is the help for Online Change"
TRAN_QUERY_STATUS_LIST_PSTO_NAME
    "Process Stop"
10 TRAN_QUERY_STATUS_LIST_PSTO_HELP
    "This is the help for Process Stop"
TRAN_QUERY_STATUS_LIST_PUR_NAME
    "Purge"
15 TRAN_QUERY_STATUS_LIST_PUR_HELP
    "This is the help for Purge"
TRAN_QUERY_STATUS_LIST_QERR_NAME
    "Queue Error"
20 TRAN_QUERY_STATUS_LIST_QERR_HELP
    "This is the help for Queue Error"
TRAN_QUERY_STATUS_LIST_SPND_NAME
    "Suspended"
25 TRAN_QUERY_STATUS_LIST_SPND_HELP
    "This is the help for Suspended"
TRAN_QUERY_STATUS_LIST_STO_NAME
    "Stopped"
30 TRAN_QUERY_STATUS_LIST_STO_HELP
    "This is the help for Stopped"
TRAN_QUERY_STATUS_LIST_TRC_NAME
    "Trace"
35 TRAN_QUERY_STATUS_LIST_TRC_HELP
    "This is the help for Trace"
TRAN_QUERY_STATUS_LIST_USTO_NAME
    "User Stop"
40 TRAN_QUERY_STATUS_LIST_USTO_HELP
    "This is the help for User Stop"

TRAN_QUERY_QCNT_NAME
    "QCount"
45 TRAN_QUERY_QCNT_STEP_TITLE
    "Specify the queue count of transactions to display"
TRAN_QUERY_QCNT_STEP_TEXT
    "Use this page to filter transactions by queue count.
50     Choose the comparison operator from the list and then type a
        number in the entry field. The following operators apply:

```

55

<TABLE>

<TR><TD>

GT > Greater than

LT < Less than

EQ = Equal to

</TD>

<TD>

GE >= Greater than or equal to

LE <= Less than or equal to

NE <> Not equal to

</TD>

</TABLE>"

TRAN_QUERY_QCNT_CONTROLS_TITLE

"Queue Count"

TRAN_QUERY_QCNT_OPERATOR

"Operator"

TRAN_QUERY_QCNT_NUMBER

"Number"

TRAN_QUERY_QCNT_NUMBER_GT_NAME

" > "

TRAN_QUERY_QCNT_NUMBER_LT_NAME

" < "

TRAN_QUERY_QCNT_NUMBER_EQ_NAME

" = "

TRAN_QUERY_QCNT_NUMBER_GE_NAME

" >= "

TRAN_QUERY_QCNT_NUMBER_LE_NAME

" <= "

TRAN_QUERY_QCNT_NUMBER_NE_NAME

" <> "

TRAN_QUERY_SHOW_NAME

"Show"

TRAN_QUERY_SHOW_STEP_TITLE

"Select the transaction output field to display"

TRAN_QUERY_SHOW_STEP_TEXT

"Specifies the transaction output fields to be

```

    displayed and overrides the default transaction
    output fields."
5  TRAN_QUERY_SHOW_CONTROLS_TITLE
    "Options"
    TRAN_QUERY_SHOW_CONTROLS_HELP
    "Option help"
10  TRAN_QUERY_SHOW_DEFAULT_NAME
    "Default - None"
    TRAN_QUERY_SHOW_DEFAULT_HELP
    "Show is an optional keyword. Select this to
    prevent SHOW from appearing in your command."
15  TRAN_QUERY_SHOW_ALL_NAME
    "All"
    TRAN_QUERY_SHOW_ALL_HELP
    "Selects all fields"
20  TRAN_QUERY_SHOW_GLOBAL_NAME
    "Global"
    TRAN_QUERY_SHOW_GLOBAL_HELP
    "This is the help for Show Global."
25  TRAN_QUERY_SHOW_LOCAL_NAME
    "Local"
    TRAN_QUERY_SHOW_LOCAL_HELP
    "This is the help for Show Local."
30  TRAN_QUERY_SHOW_LIST_NAME
    "Select - List"
    TRAN_QUERY_SHOW_LIST_HELP
    "Select from the list"
35  TRAN_QUERY_SHOW_LIST_CLS_NAME
    "Class"
    TRAN_QUERY_SHOW_LIST_CLS_HELP
    "This is the help for Class"
40  TRAN_QUERY_SHOW_LIST_CP_NAME
    "CP"
    TRAN_QUERY_SHOW_LIST_CP_HELP
    "Current scheduling priority"
45  TRAN_QUERY_SHOW_LIST_LCT_NAME
    "LCT"
    TRAN_QUERY_SHOW_LIST_LCT_HELP
    "Limit Count."
50  TRAN_QUERY_SHOW_LIST_LP_NAME
    "LP"

```

55

```

TRAN_QUERY_SHOW_LIST_LP_HELP
    "Limit scheduling priority"
5  TRAN_QUERY_SHOW_LIST_NP_NAME
    "NP"
TRAN_QUERY_SHOW_LIST_NP_HELP
    "Normal scheduling priority."
10 TRAN_QUERY_SHOW_LIST_PARLM_NAME
    "PARLM"
TRAN_QUERY_SHOW_LIST_PARLM_HELP
    "Parallel processing limit count, which is the maximum number of
15     messages that can currently be queued, but not yet processed,
    by each active message region currently scheduled for this
    transaction."
TRAN_QUERY_SHOW_LIST_PLCT_NAME
    "PLCT"
20 TRAN_QUERY_SHOW_LIST_PLCT_HELP
    "Processing limit count"
TRAN_QUERY_SHOW_LIST_PSBNAME_NAME
    "PSBNAME"
25 TRAN_QUERY_SHOW_LIST_PSBNAME_HELP
    "PSB name associated with the transaction."
TRAN_QUERY_SHOW_LIST_QCNT_NAME
    "QCNT"
30 TRAN_QUERY_SHOW_LIST_QCNT_HELP
    "Display the global queue count and the local queue count."
TRAN_QUERY_SHOW_LIST_RC_NAME
    "RC"
35 TRAN_QUERY_SHOW_LIST_RC_HELP
    "Number of regions the transaction is currently scheduled in."
TRAN_QUERY_SHOW_LIST_SEGNO_NAME
    "SEGNO"
40 TRAN_QUERY_SHOW_LIST_SEGNO_HELP
    "Application program output segment limit allowed in message
    queues for each GU call."
45 TRAN_QUERY_SHOW_LIST_SEGSZ_NAME
    "SEGSZ"
TRAN_QUERY_SHOW_LIST_SEGSZ_HELP
    "Application program output segment size limit allowed in the
50     message queues for each GU call."
TRAN_QUERY_SHOW_LIST_STATUS_NAME
    "STATUS"

```

55

TRAN_QUERY_SHOW_LIST_STATUS_HELP
 "Global and local transaction status."

5 TRAN_QUERY_ROUTE_NAME
 "Route"

TRAN_QUERY_ROUTE_STEP_TITLE
 "Specify the systems that will receive this command"

10 TRAN_QUERY_ROUTE_STEP_TEXT
 "This is the explanation of how to complete this step.
 It includes info on what the terms mean and how to use the
 controls and any available help."

15 TRAN_QUERY_ROUTE_CONTROLS_TITLE
 "Modular unit id"

TRAN_UPDATE_NAME
 "Update"

20 TRAN_UPDATE_NAME_NAME
 "Name"

TRAN_UPDATE_NAME_STEP_TITLE
 "Name transactions to update"

25 TRAN_UPDATE_NAME_STEP_TEXT
 "This page lists the names of the transactions that you will
 update. Type a transaction name in the entry field and
 press Enter to add it to the list. You can also select
 30 previous transactions from the drop-down list and use the
 wildcard (*) character to display all or filter transactions.
 For example, ABC* will update all transactions starting with
 ABC."

35 TRAN_UPDATE_NAME_CONTROLS_TITLE
 "Transaction names (*)"

TRAN_UPDATE_CLASS_NAME
 "Class"

40 TRAN_UPDATE_CLASS_STEP_TITLE
 "Specify the class number of transactions to update"

TRAN_UPDATE_CLASS_STEP_TEXT
 "This page lists the class numbers of the transactions that you
 will update. Type a class number in the entry field and
 45 press Enter to add it to the list. You can also select
 previous class numbers from the drop-down list."

TRAN_UPDATE_CLASS_CONTROLS_TITLE
 "Class number"

50 TRAN_UPDATE_START_NAME
 "Start"

55

TRAN_UPDATE_START_STEP_TITLE
 "Select transaction functions to stop"

5 TRAN_UPDATE_START_STEP_TEXT
 "This page lists transaction functions that you can start.
 The start will apply to the transactions that you selected
 in the Name and Class steps. To see a detailed explanation
 of an option, move the cursor to that option."

10 TRAN_UPDATE_START_CONTROLS_TITLE
 "Options"

TRAN_UPDATE_START_CONTROLS_HELP
 "Option help"

15 TRAN_UPDATE_START_DEFAULT_NAME
 "Default - None"

TRAN_UPDATE_START_DEFAULT_HELP
 "This is the default -- do not start any functions"

20 TRAN_UPDATE_START_ALL_NAME
 "All"

TRAN_UPDATE_START_ALL_HELP
 "Start all of the transaction functions."

25 TRAN_UPDATE_START_LIST_NAME
 "Select - List"

TRAN_UPDATE_START_LIST_HELP
 "Select the start options from the list."

30 TRAN_UPDATE_START_LIST_IN_NAME
 "Input"

TRAN_UPDATE_START_LIST_IN_HELP
 "This is the help for Input."

35 TRAN_UPDATE_START_LIST_OUT_NAME
 "Output"

TRAN_UPDATE_START_LIST_OUT_HELP
 "This is the help for Output"

40 TRAN_UPDATE_START_LIST_QUE_NAME
 "Queueing"

TRAN_UPDATE_START_LIST_QUE_HELP
 "This is the help for Queueing"

45 TRAN_UPDATE_START_LIST_SCHED_NAME
 "Scheduling"

TRAN_UPDATE_START_LIST_SCHED_HELP
 "This is the help for Scheduling"

50 TRAN_UPDATE_START_LIST_SPNDQ_NAME
 "Suspend Queue"

55


```

TRAN_UPDATE_START_LIST_SPNDQ_HELP
    "This is the help for Suspend Queue"
TRAN_UPDATE_START_LIST_TRC_NAME
    "Trace"
TRAN_UPDATE_START_LIST_TRC_HELP
    "This is the help for Trace"
TRAN_UPDATE_STOP_NAME
    "Stop"
TRAN_UPDATE_STOP_STEP_TITLE
    "Select transaction functions to stop"
TRAN_UPDATE_STOP_STEP_TEXT
    "This page lists transaction functions that you can stop.
    The stop will apply to the transactions that you selected
    in the Name and Class steps. To see a detailed explanation
    of an option, move the cursor to that option."
TRAN_UPDATE_STOP_CONTROLS_TITLE
    "Options"
TRAN_UPDATE_STOP_CONTROLS_HELP
    "Option help"
TRAN_UPDATE_STOP_DEFAULT_NAME
    "Default - None"
TRAN_UPDATE_STOP_DEFAULT_HELP
    "This is the default -- do not stop any functions"
TRAN_UPDATE_STOP_ALL_NAME
    "All"
TRAN_UPDATE_STOP_ALL_HELP
    "Stop all of the transaction functions."
TRAN_UPDATE_STOP_LIST_NAME
    "Select - List"
TRAN_UPDATE_STOP_LIST_HELP
    "Select the stop options from the list."
TRAN_UPDATE_STOP_LIST_IN_NAME
    "Input"
TRAN_UPDATE_STOP_LIST_IN_HELP
    "This is the help for Input"
TRAN_UPDATE_STOP_LIST_OUT_NAME
    "Output"
TRAN_UPDATE_STOP_LIST_OUT_HELP
    "This is the help for Output"
TRAN_UPDATE_STOP_LIST_QUE_NAME
    "Queueing"

```

TRAN_UPDATE_STOP_LIST_QUE_HELP
 "This is the help for Queueing"
 TRAN_UPDATE_STOP_LIST_SCHED_NAME
 "Scheduling"
 TRAN_UPDATE_STOP_LIST_SCHED_HELP
 "This is the help for Scheduling"
 TRAN_UPDATE_STOP_LIST_TRC_NAME
 "Trace"
 TRAN_UPDATE_STOP_LIST_TRC_HELP
 "This is the help for Trace"
 TRAN_UPDATE_SET_NAME
 "Set"
 TRAN_UPDATE_SET_STEP_TITLE
 "Specify the transaction attributes you want to change"
 TRAN_UPDATE_SET_STEP_TEXT
 "This page lists transaction attributes that you can change.
 Select the attribute, then type the new value in the entry
 field. To see a detailed explanation
 of an attribute, move the cursor to that option."
 TRAN_UPDATE_SET_CONTROLS_TITLE
 "Options"
 TRAN_UPDATE_SET_CONTROLS_HELP
 "Option help"
 TRAN_UPDATE_SET_LIST_CLASS_NAME
 "Class"
 TRAN_UPDATE_SET_LIST_CLASS_HELP
 "This is the help for Class"
 TRAN_UPDATE_SET_LIST_CPRI_NAME
 "Current Priority"
 TRAN_UPDATE_SET_LIST_CPRI_HELP
 "This is the help for Current Priority"
 TRAN_UPDATE_SET_LIST_LMCT_NAME
 "Limit Count"
 TRAN_UPDATE_SET_LIST_LMCT_HELP
 "This is the help for Limit Count"
 TRAN_UPDATE_SET_LIST_LPRI_NAME
 "Limit Priority"
 TRAN_UPDATE_SET_LIST_LPRI_HELP
 "This is the help for Limit Priority"
 TRAN_UPDATE_SET_LIST_MAXRGNS_NAME
 "Max Regions"

TRAN_UPDATE_SET_LIST_MAXRGNS_HELP
 "This is the help for Max Regions"
 5 TRAN_UPDATE_SET_LIST_NPRI_NAME
 "Normal Priority"
 TRAN_UPDATE_SET_LIST_NPRI_HELP
 "This is the help for Normal Priority"
 10 TRAN_UPDATE_SET_LIST_PARLIM_NAME
 "Parallel Limit Count"
 TRAN_UPDATE_SET_LIST_PARLIM_HELP
 "This is the help for Parallel Limit Count"
 15 TRAN_UPDATE_SET_LIST_PLMCT_NAME
 "Processing Limit Count"
 TRAN_UPDATE_SET_LIST_PLMCT_HELP
 "This is the help for Processing Limit Count"
 20 TRAN_UPDATE_SET_LIST_SEGNO_NAME
 "Segment Number"
 25 TRAN_UPDATE_SET_LIST_SEGNO_HELP
 "This is the help for Segment Number"
 TRAN_UPDATE_SET_LIST_SEGSIZE_NAME
 "Segment Size"
 30 TRAN_UPDATE_SET_LIST_SEGSIZE_HELP
 "This is the help for Segment Size"
 TRAN_UPDATE_ROUTE_NAME
 35 "Route"
 TRAN_UPDATE_ROUTE_STEP_TITLE
 "Specify the systems that will receive this command"
 TRAN_UPDATE_ROUTE_STEP_TEXT
 40 "This is the explanation of how to complete this step.
 It includes info on what the terms mean and how to use the
 controls and any available help."
 45 TRAN_UPDATE_ROUTE_CONTROLS_TITLE
 "Modular unit id"

Claims

1. A method for generating a graphical user interface from a command syntax, for managing at least one computer system, comprising:

(a) processing at least one eXtensible Markup Language (XML) file (112) at a user interface program (114) executed by a computer, wherein the XML file represents a command syntax for managing the at least one

computer system; and

(b) generating at least one dialog from the XML file for display to an operator, wherein the at least one dialog is used to create at least one command in the user interface program for managing the at least one computer system.

2. The method of claim 1, wherein the XML files are selected from a group comprising an XML document file, XML schema file, and text description file.
3. The method of claim 2, wherein the XML document, XML schema, and text description file are maintained on a server (106) and are downloaded to the computer as required.
4. The method of claim 3, wherein the XML document, XML schema, and text description file are modified on the server as the command syntax changes.
5. The method of claim 2, 3 or 4, wherein the XML document, XML schema, and text description file represent a latest version of the command syntax.
6. The method of any of claims 2 to 5, wherein the user interface program processes the XML document using the XML schema to identify resource types, verbs, and keywords therein.
7. The method of any of claims 2 to 6, wherein the text description file provides at least one text description that is used by the user interface program to construct the at least one dialog.
8. The method of any of claims 2 to 7, wherein the text description file comprises at least one text description that is represented by tags that comprise concatenations of nested XML tag names in the XML document.
9. The method of any of claims 2 to 8, wherein the XML document comprises one or more tags selected from a group comprising:
 - a root tag that is a global tag that provides a wrapper for all other tags,
 - a resource tag that is used to define a resource,
 - a verb tag that is used to define a verb,
 - a keyword tag that is used to define a keyword,
 - a list tag that is used within the keyword tag to define a list of options,
 - a compare tag that is used within the keyword tag to define a comparison between an operator and a variable,
 - an option tag that is used within the list tag to define optional selections for a keyword,
 - a variable tag that is used within the keyword and list tags to define a variable, and
 - an operator tag that is used within the compare tag to define an operator for a list of operators.
10. The method of any preceding claim, wherein the at least one dialog comprises a Wizard for use by the operator in generating commands for the at least one computer system.
11. The method of claim 10, wherein the Wizard displays a step-by-step series of dialogs for guiding the operator in the use of the command syntax.
12. The method of any preceding claim, wherein the commands are submitted to multiple computer systems.
13. The method of any preceding claim, wherein the user interface program is not sensitive to an individual format of the command syntax that is used to manage the at least one computer system.

14. A computer program product comprising computer program code stored on a computer readable storage medium, which when executed on a computer, performs the method of any preceding claim.

5 15. A system for generating a graphical user interface from a command syntax for managing at least one computer system, comprising:

(a) a computer; and

10 (b) a user interface program (114), executed by the computer, for processing at least one eXtensible Markup Language (XML) file (112) at a user interface program executed by a computer, wherein the XML file represents a command syntax for managing the at least one computer system, and for generating at least one dialog from the XML file for display to an operator, wherein the at least one dialog is used to create at least one command in the user interface program for managing the at least one computer system.

15

20

25

30

35

40

45

50

55

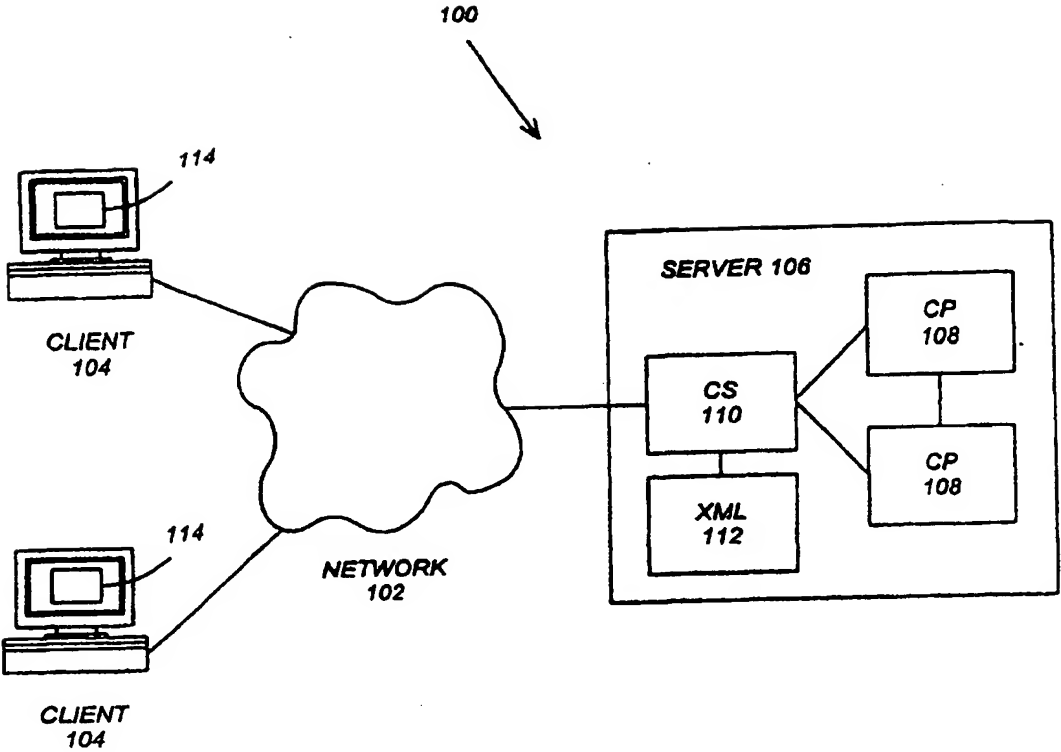


FIG. 1

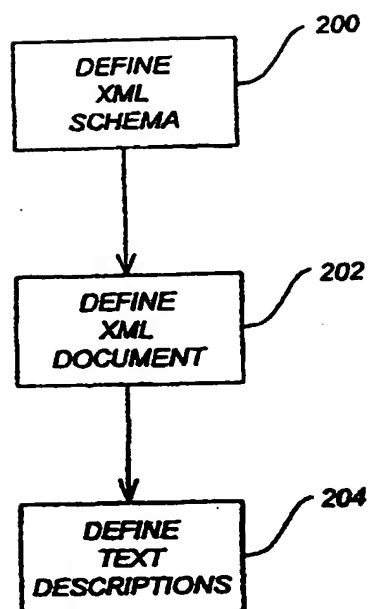


FIG. 2

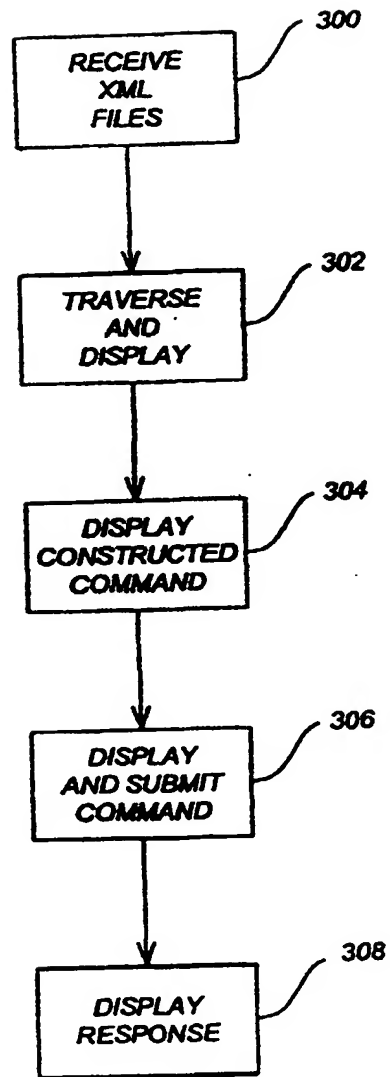


FIG. 3

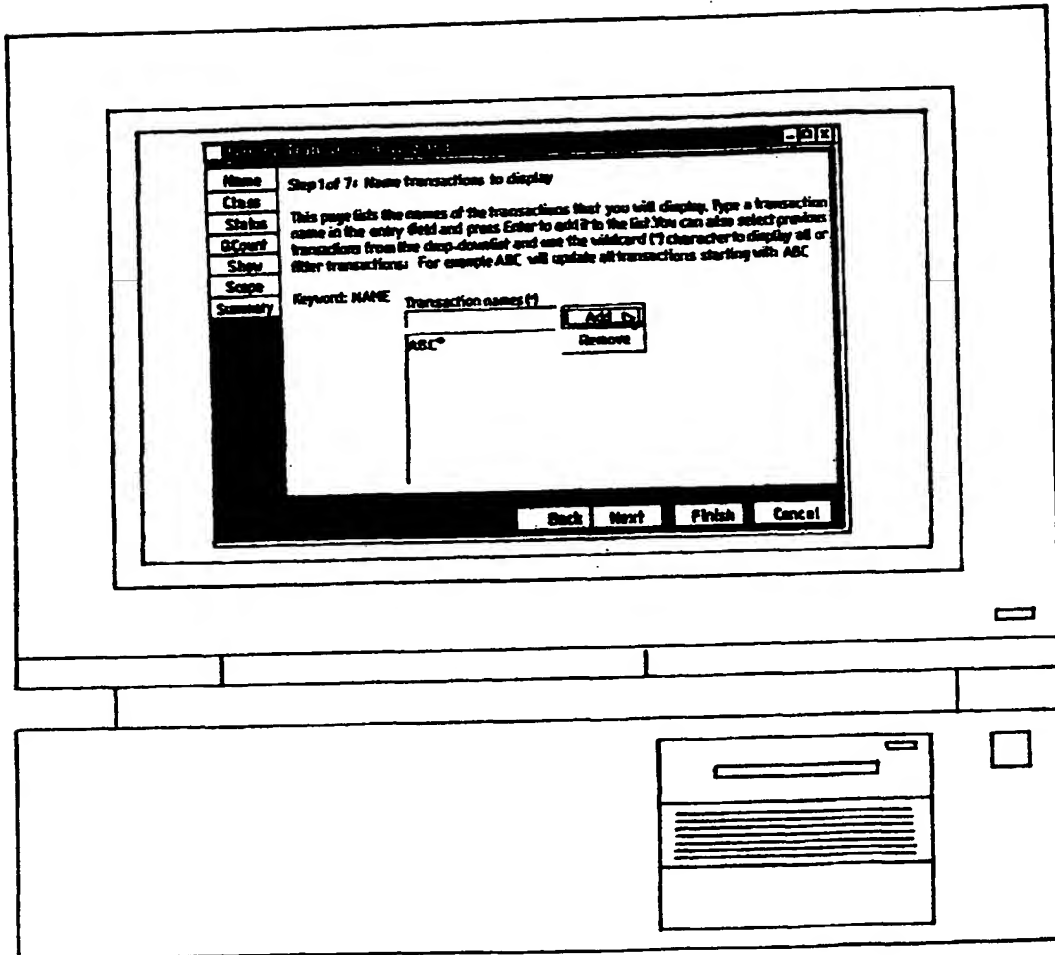


FIG. 4A

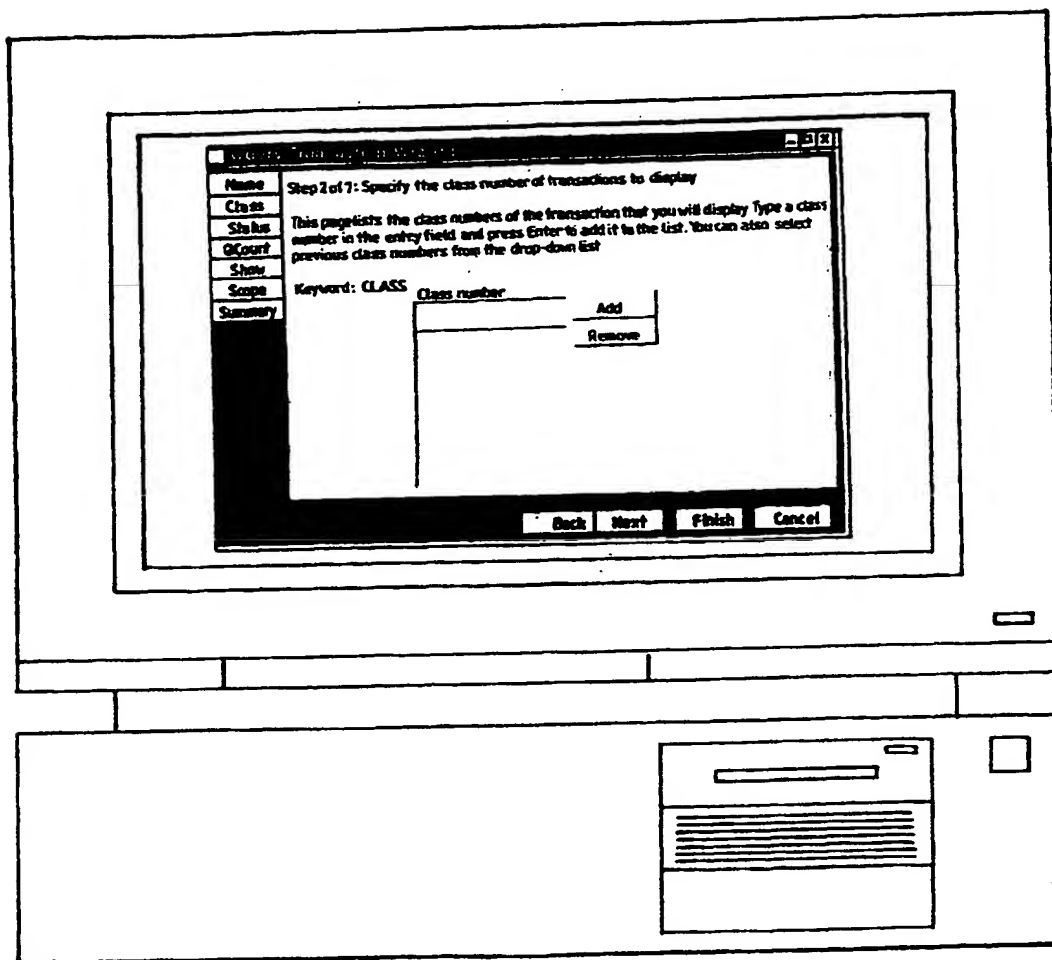


FIG. 4B

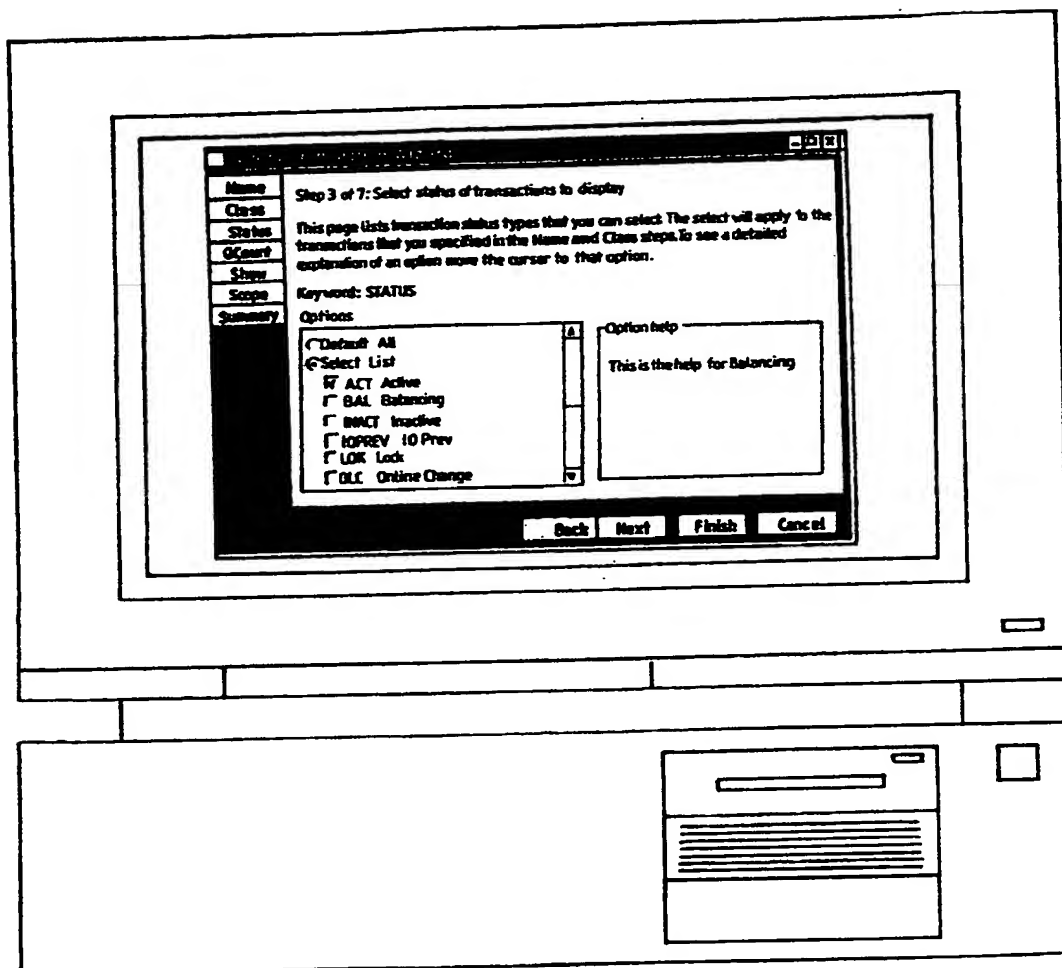


FIG. 4C

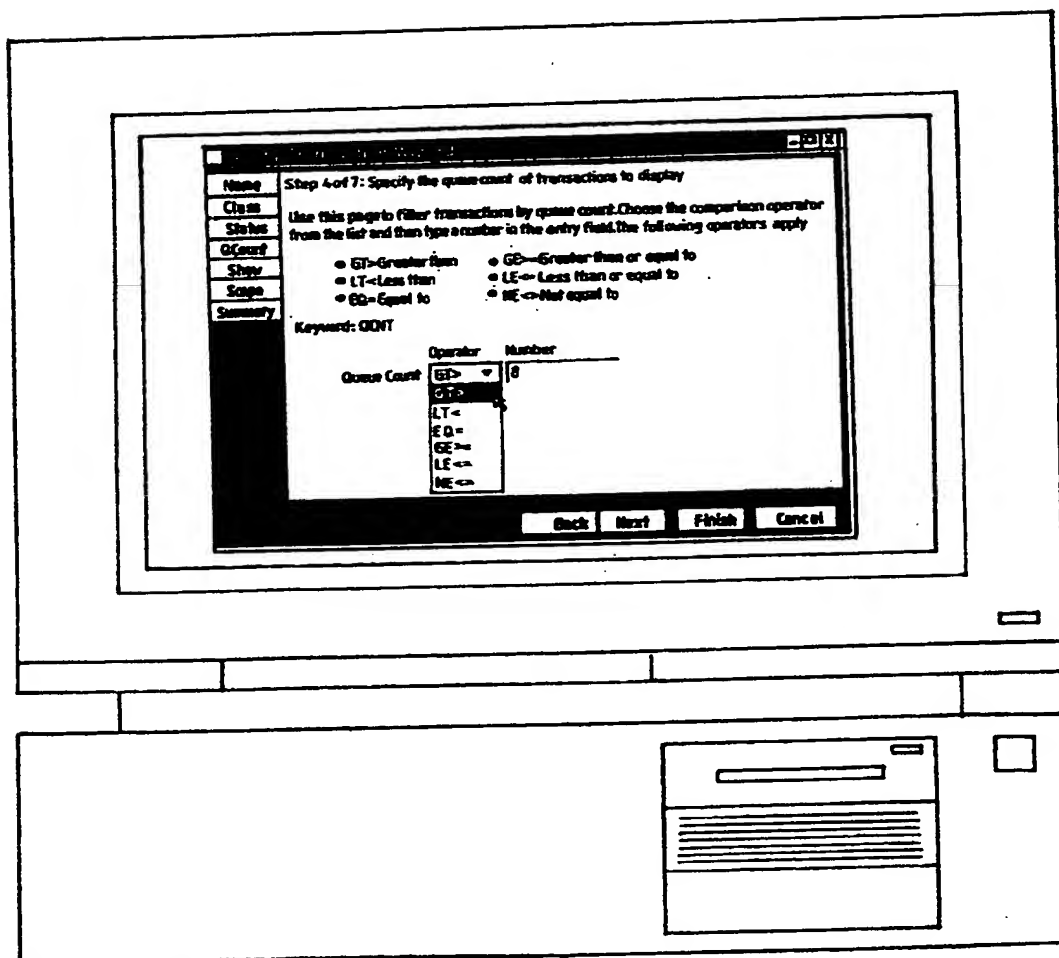


FIG. 4D

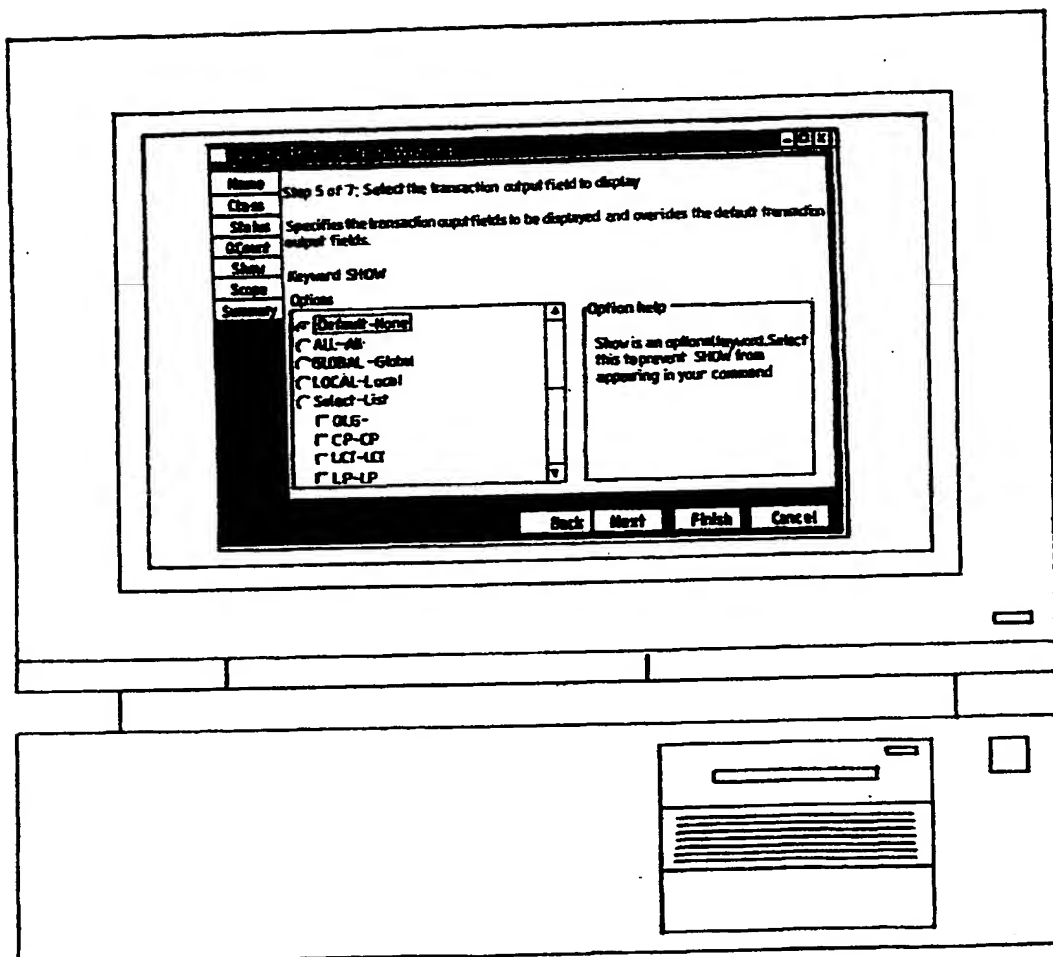


FIG. 4E

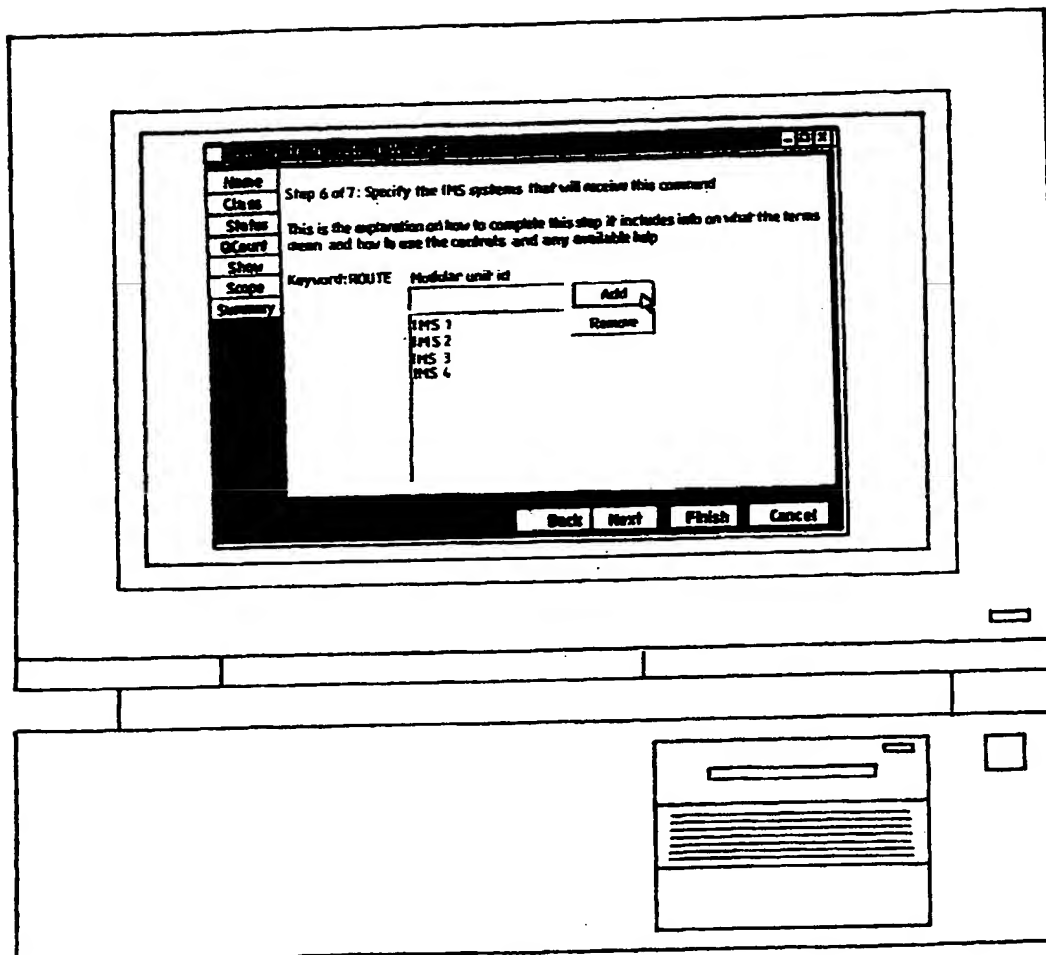


FIG. 4F

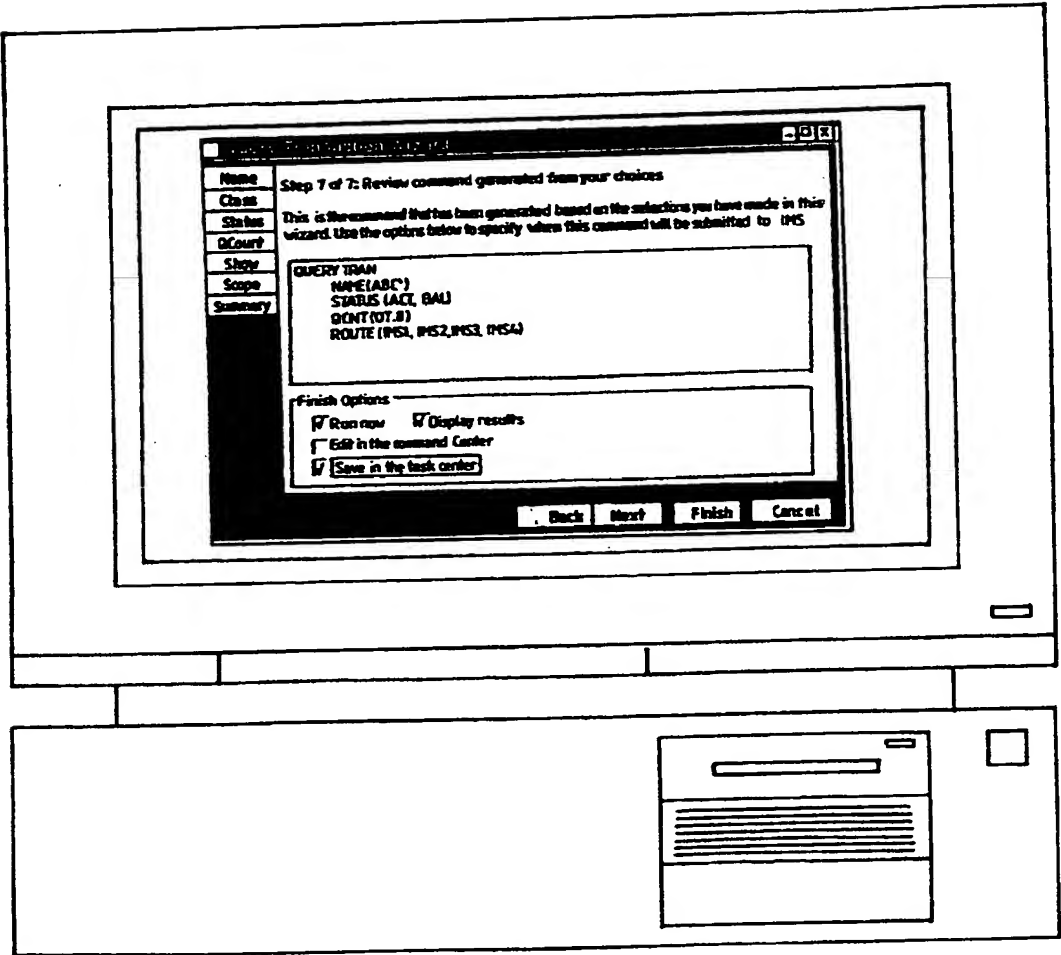


FIG. 4G

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☒ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☒ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.